|  |  |
| --- | --- |
| **Assignment:** | PS 08, Problem 1 |
| **Name:** | Harith Kolaganti, hkolagan@purdue.edu |
| **Team-ID** | 005-12 |
| **Contributor(s):** |  |

# Logic Truth Table

Individual

Construct and complete the truth tables as required in the instructions below. Remember: **to receive full credit, you must show all intermediate steps needed to reach the final answer for each location in each truth table—even if you see opportunities for short cuts**.

As you work through the problems, note that for

**Parts 3.c – 3.d:** You do not need to complete cells that are greyed out.

**Parts 4.e – 4.f:** You must decide which cells are necessary to evaluate a statement

(**Note:** each truth table in this assignment is on its own page)

Complete the truth table for the logic statement

**(B & ~A) & xor(A,B)**

|  |  |  |
| --- | --- | --- |
|  | B = 0 | B = 1 |
| A = 0 | (0 & ~0) & xor(0,0)  (0 & 1) & 0  0 & 0  0 | (1 & ~0) & xor(0,1)  (1 & 1) & 1  1 & 1  1 |
| A = 1 | (0 & ~1) & xor(1,0)  (0 & 0) & 1  0 & 1  0 | (1 & ~1) & xor(1,1)  (1 & 0) & 0  0 & 0  0 |

Complete the truth table for the logic statement

**~(L & M == ~N) | N & ~M > L**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | L=0 | | L=1 | |
|  | M=0 | M=1 | M=0 | M=1 |
| N=0 |  | **~(0 & 1 == ~0) | 0 & ~1 >0**  **~(0) | 0 & 0 > 0**  **1 | 0 & 0 > 0**  **1 | 0 > 0**  **1 > 0**  **1** | **~(1 & 0 == ~0) | 0 & ~0 > 1**  **~(0==1) | 0 & ~0 > 1**  **~(0) | 0 & 1 > 1**  **1 | 0 & 1 > 1**  **1 | 0 > 1**  **1 | 0**  **1** | **~(1 & 1 == ~0) | 0 & ~1 > 1**  **~(1 == 1) | 0 & 0 > 1**  **~1 | 0 & 0 >1**  **0 | 0 & 0 > 1**  **1 & 0 > 1**  **0 > 1**  **0** |
| N=1 | **~(0 & 0 == ~1) | 1 & ~0 > 0**  **~(0 ==0) | 1 & 1 > 0**  **0 | 1 & 1 > 0**  **1 & 1 > 0**  **1 > 0**  **1** |  | **~(1 & 0 == ~1) | 1 & ~0 > 1**  **~(0==0) | 1 & 1 > 1**  **~1 | 1 & 1 > 1**  **0 | 1 > 1**  **1 > 1**  **0** |  |

Complete the truth table for the logic statement.

**(~xor(F,G) & H) < K == G**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | F=0 | | F=1 | |
|  |  | G=0 | G=1 | G=0 | G=1 |
| H=0 | K=0 | **(~xor(0,0) & 0) < 0 == 0**  **~(0 & 0) < 0 == 0**  **~0 < 0 == 0**  **1 < 0 == 0**  **0 == 0**  **1** | **(~xor(0,1) & 0) < 0 == 1**  **(~1 & 0) < 0 == 1**  **(0 & 0) < 0 == 1**  **0 & 0 == 1**  **0 == 1**  **0** |  | **(~xor(1,1) & 0) < 0 == 1**  **(~0 & 0) < 0 == 1**  **(1 & 0) < 0 == 1**  **0 < 0 == 1**  **0 == 1**  **0** |
| K=1 |  | **(~xor(0,1) & 0) < 1 == 1**  **(~1 & 0) < 1 == 1**  **(0 & 0) < 1 == 1**  **0 < 1 == 1**  **1 == 1**  **1** |  | **(~xor(1,1) & 0) < 1 == 1**  **(~0 & 0) < 1 == 1**  **0 < 1 == 1**  **1 == 1**  **1** |
| H=1 | K=0 |  | **(~xor(0,1) & 1) < 0 == 1**  **(~1 & 1) < 0 == 1**  **(0 & 1) < 0 == 1**  **0 < 0 == 1**  **0 == 1**  **0** | **(~xor(1,0) & 1) < 0 == 0**  **(~1 & 1) < 0 == 0**  **(0 & 1) < 0 == 0**  **0 < 0 == 0**  **0 == 0**  **1** |  |
| K=1 | **(~xor(0,0) & 1) < 1 == 0**  **(~0 & 1) < 1 == 0**  **(1 & 1) < 1 == 0**  **1 < 1 == 0**  **0 == 0**  **1** |  |  | **(~xor(1,1) & 1) < 1 == 1**  **(~0 & 1) < 1 == 1**  **1 < 1 == 1**  **0 == 1**  **0** |

Construct the truth table and then complete it for the logic statement

**( E | C) & D | (D&E)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | E=0 | | E=1 | |
|  | C=0 | C=1 | C=0 | C=1 |
| D=0 | **( 0 | 0) & 0 | (0&0)**  **0 & 0 | 0**  **0 | 0**  **0** | **( 0 | 1) & 0 | (0&0)**  **1 & 0 | 0**  **0 | 0**  **0** | **( 0 | 0) & 0 | (0&1)**  **0 & 0 | 0**  **0 | 0**  **0** | **( 0 | 1) & 0 | (0&1)**  **1 & 0 | 0**  **0 | 0**  **0** |
| D=1 | **( 0 | 0) & 1 | (1&0)**  **0 & 1 | 0**  **0 | 0**  **0** | **( 0 | 1) & 1 | (1&0)**  **1 & 1 | 0**  **1 | 0**  **1** | **( 1 | 0) & 1 | (1&1)**  **1 & 1 | 1**  **1 | 1**  **1** | **( 1 | 1) & 1 | (1&1)**  **1 & 1 | 1**  **1 | 1**  **1** |

## Part 4.e

Translate the English logic statement:

The statement is true if both P and Q are true. The statement is true if both P and Q are false. The statement is true if R is true. All other conditions are false.

**T/F pattern identification from English statement**

Add the variables to the table. Then, fill with 1 and 0 according to your interpretation of the English statement.

|  |  |  | |  | |
| --- | --- | --- | --- | --- | --- |
|  |  | P = 0 | P = 1 |  |  |
| R = 1 | Q = 1 | 1 | 1 |  |  |
| Q = 0 | 1 | 1 |  |  |
| R = 0 | Q = 1 | 0 | 1 |  |  |
| Q = 0 | 1 | 0 |  |  |

Use the pattern above to write a MATLAB logic statement that produces the same outputs

|  |  |
| --- | --- |
| **Logical Statement** | ~xor(P,Q) | R |

**T/F pattern verification using MATLAB logic statement**

Add the variables in the same manner as the pattern table. Fill this table with complete work using the logical statement you have written above. Intermediate steps must be provided in each cell of the table — even if you see opportunities for short cuts.

|  |  |  | |  | |
| --- | --- | --- | --- | --- | --- |
|  |  | P = 0 | P = 1 |  |  |
| R = 1 | Q = 1 | ~xor(0,1) | 1  0 | 1  1 | ~xor(1,1) | 1  1 | 1  1 |  |  |
| Q = 0 | ~xor(0,0) | 1  1 | 1  1 | ~xor(1,0) | 1  0 | 1  1 |  |  |
| R = 0 | Q = 1 | ~xor(0,1) | 0  0 | 0  0 | ~xor(1,1) | 0  1 | 0  0 |  |  |
| Q = 0 | ~xor(0,0) | 0  1 | 0  1 | ~xor(1,0) | 0  0 | 0  0 |  |  |

## Part 4.f

Translate the English logic statement:

The statement is true if one of S or T is true, or the statement is true if S is true and V is false.

**T/F pattern identification from English statement**

Add the variables to the table. Then, fill with 1 and 0 according to your interpretation of the English statement.

|  |  |  | |  | |
| --- | --- | --- | --- | --- | --- |
|  |  | S = 0 | S = 1 |  |  |
| V = 0 | T = 0 | 0 | 1 |  |  |
| T = 1 | 1 | 1 |  |  |
| V = 1 | T = 0 | 0 | 1 |  |  |
| T = 1 | 1 | 0 |  |  |

Use the pattern above to write a MATLAB logic statement that produces the same outputs

|  |  |
| --- | --- |
| **Logical Statement** | xor(S,T) | (S & ~V) |

**T/F pattern verification using MATLAB logic statement**

Add the variables in the same manner as the pattern table. Fill this table with complete work using the logical statement you have written above. Intermediate steps must be provided in each cell of the table — even if you see opportunities for short cuts.

|  |  |  | |  | |
| --- | --- | --- | --- | --- | --- |
|  |  | S = 0 | S = 1 |  |  |
| V = 0 | T = 0 | xor(0,0) | (0 & ~0)  0 | (0 & 1)  0 | 0  0 | xor(1,0) | (1 & ~0)  1 | (1 & 1)  1 | 1  1 |  |  |
| T = 1 | xor(0,1) | (0 & ~0)  1 | (0 & 1)  1 | 0  1 | xor(1,1) | (1 & ~0)  0 | (1 & 1)  0 | 1  1 |  |  |
| V = 1 | T = 0 | xor(0,0) | (0 & ~1)  0 | (0 & 0)  0 | 0  0 | xor(1,0) | (1 & ~1)  1 | (1 & 0)  1 | 0  1 |  |  |
| T = 1 | xor(0,1) | (0 & ~1)  1 | (0 & 0)  1| 0  1 | xor(1,1) | (1 & ~1)  0 | (1 & 0)  0 | 0  0 |  |  |